

NASA Contractor Report 194953

ICASE Report No. 94-62

AD-A284 064

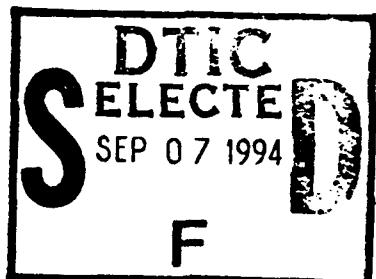


(1)



ICASE

AGGLOMERATION MULTIGRID FOR VISCOUS TURBULENT FLOWS



D.J. Mavriplis
V. Venkatakrishnan

308 94-29067



Contract NAS1-19480
July 1994

This document has been approved
for public release and sale; its
distribution is unlimited

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA 23681-0001

DTIC QUALITY INSPECTED 3



Operated by Universities Space Research Association

94 9 06 114

AGGLOMERATION MULTIGRID FOR VISCOUS TURBULENT FLOWS

D. J. Mavriplis

and

V. Venkatakrishnan

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution	
Avail	
Dist	

A-1

ABSTRACT

Agglomeration multigrid, which has been demonstrated as an efficient and automatic technique for the solution of the Euler equations on unstructured meshes, is extended to viscous turbulent flows. For diffusion terms, coarse grid discretizations are not possible, and more accurate grid transfer operators are required as well. A Galerkin coarse grid operator construction and an implicit prolongation operator are proposed. Their suitability is evaluated by examining their effect on the solution of Laplace's equation. The resulting strategy is employed to solve the Reynolds-averaged Navier-Stokes equations for aerodynamic flows. Convergence rates comparable to those obtained by a previously developed non-nested mesh multigrid approach are demonstrated, and suggestions for further improvements are given.

This research was supported under the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681.

1. INTRODUCTION

Over the last several years, multigrid techniques for unstructured meshes have been demonstrated to provide an efficient solution mechanism for steady-state flows, both in two dimensions [1,2,3], and in three dimensions [4,5,6]. When operating on unstructured meshes, the main difficulty associated with the use of a multigrid algorithm lies in the generation of the coarser levels. For structured grids, coarse levels are generated by removing every nth point in each coordinate direction, and using the remaining subset of points as the basis for a coarser structured grid.

For unstructured meshes, various approaches have been attempted. One approach begins with the coarse mesh definition, and generates finer nested levels by repeatedly subdividing the coarse grid cells [7,8]. While this method enables the use of simple inter-grid transfer operators, the main drawback is the dependence of the fine grid point distribution on the coarse levels. Since the fine grid uniquely affects the solution accuracy in a multigrid algorithm, optimization of the fine grid for accuracy and optimization of the coarse grid for speed of convergence can often result in conflicting requirements. Furthermore, the initial coarse grid may not be coarse enough to realize the full potential benefit of a multigrid algorithm.

One of the most successful strategies has been the use of non-nested coarse and fine levels [1,4,5,6]. In this approach, coarse grid levels are generated independently from the finer levels using any given grid generation strategy. Flow variables, residuals, and corrections are transferred back and forth between the various grid levels in a multigrid cycle using linear interpolation. Since the levels are generated independently from one another, the coarse grids are not nested with the fine grids, and generally do not even contain common points with the fine levels. Since the relationship between the various grid levels does not change throughout the multigrid convergence process, the patterns for inter-grid interpolation can be computed in a pre-processing phase using efficient graph-traversal search techniques.

A more automated but somewhat less flexible variant of this technique operates on coarse grids which are non-nested, but which are formed from subsets of the fine grid points. One approach consists of selecting fine grid point subsets and retriangulating these points using a Delaunay triangulation algorithm [9]. Although the coarse grid points are contained in the fine grid, the elements are generally not nested, since the triangulations (i.e. connectivity of the points) is recomputed on each level.

However, all these approaches share a common problem, that is the generation of coarse unstructured grids. For complex geometries, it is often difficult to generate a coarse grid which preserves the original geometry. In other words, there comes a point where certain features in the geometry become finer than the desired grid resolution, and a coarser grid may either not be possible, or may alter the geometry or even the topology of the geometry. From a practical point of view, the need to generate multiple meshes for a single solution places an excessive burden on the user, particularly for three dimensional computations. An alternative to the above methods, which circumvents this problem, is the agglomeration multigrid strategy. In this method, coarse grids are formed by fusing together fine grid cells. On unstructured triangular or tetrahedral grids, the resulting coarse grids contain polygonal or polyhedral cells which are no longer simple triangles or tetrahedra. A method for discretizing and solving the governing equations on these coarse grids must therefore be devised. Agglomeration multigrid was originally introduced by Lallemand et al. [2] for vertex schemes, and has been developed apparently independently for cell-centered schemes by Smith [3]. However, early published results on agglomeration multigrid failed to demonstrate efficiency levels comparable to those

of the non-nested unstructured multigrid methods and of regular structured multigrid methods. More recently, it has been shown how agglomeration multigrid strategies can be made competitive with other multigrid strategies for the two and three-dimensional Euler equations, both in terms of convergence rates, and in terms of complexity [10,11,12]. Agglomeration methods applied to diffusion problems have been reported by Koobus et al. [13].

Another method which avoids the generation of coarse grids is the algebraic multigrid approach [14]. Algebraic multigrid rates on the matrix of the discrete operator, rather than on the grid of the discretization. Nearest neighbor discretization on an unstructured grid, the graph of the discrete operator matrix is identical to the graph of the grid [4], thus analogies between algebraic multigrid methods and geometric agglomeration multigrid strategies can be drawn. Algebraic multigrid methods consist of a setup phase, and a solution phase. In the setup phase, the coarse levels are constructed, the inter-level transfer operators (restriction and prolongation) are determined, and the coarse level operators are constructed. These elements are then used to solve the fine level equations in a multi-level cycling procedure. The coarse levels are formed by considering subsets of the fine level variables. In grid terminology, this means that the fine grid is traversed, and selected fine grid points are deleted, thus leaving a smaller subset of points for the next level. The prolongation operator P is determined by requiring that the prolonged corrections be algebraically smooth on the fine level. If the fine grid discrete equations are written as

$$A u = f \quad (1)$$

Mathematically, the requirement of smooth corrections is characterized by

$$A e \approx 0 \quad (2)$$

where e represents the prolonged corrections. The restriction operator R is usually taken as the transpose of the prolongation operator. The coarse equations are then written as

$$\bar{A} \bar{u} = \bar{f} \quad (3)$$

where

$$\bar{A} = R A P \quad (4)$$

and

$$\bar{f} = R f \quad (5)$$

This construction is in contrast to other multigrid methods where the coarse grid operator is formed by rediscretizing the governing equations on the new coarser level. This is often referred to as a Galerkin coarse grid operator, since it can be shown that if A minimizes a functional over the set of functions spanned on the fine level, then RAP minimizes the same functional over the set of functions spanned on the coarser level [15]. Algebraic multigrid methods have been demonstrated successfully for various types of problems. However, in their simplest form, they are limited to linear problems, and may fail for more complex problems, such as systems of equations, where the block structure of the operator matrix may not be automatically recognized by the algorithm. Their use in computational fluid dynamics problems has thus been limited.

The present paper proposes to extend the previously developed agglomeration strategies to the Navier-Stokes equations. This extension is not straight forward and requires a significant re-evaluation of the role of agglomeration. The main difficulties stem from the

discretization of the governing equations on the coarse polygonal meshes. For the inviscid terms, a simple control volume analysis can be used to derive discrete equations on arbitrary polygons or polyhedra. However, for the viscous terms, (or even for a simple Laplacian), the discretization on polygonal meshes is not obvious, since this usually requires the computation of gradients as an intermediate step. The basic strategy is developed by considering the solution of a Laplace equation using agglomeration multigrid. We draw heavily on the ideas of the algebraic multigrid method [14]. We compare the agglomeration performance with that of the overset grid method [1], and attempt to demonstrate and improve the elements of the algorithm which contribute to non-optimal convergence rates. In doing so, we recover with further justification some of the results reported in [13].

In the following section, we describe the agglomeration strategy for constructing coarse grid levels. In section 3, we formulate the prolongation, restriction and coarse grid operators, based on the performance of a Laplace equation solver, and in section 4, we extend these techniques to the two-dimensional Reynolds averaged Navier-Stokes equations.

2. AGGLOMERATION STRATEGIES

The coarse grids for use in our multigrid procedure are derived directly from the fine grid through fusion (agglomeration) of control volumes. This agglomeration is accomplished by using a greedy-type frontal algorithm and is done in such a way that the complexity, which is proportional to the number of edges, goes down by nearly a constant factor (4 in 2-d and 8 in 3-d) when moving from a fine to a coarse grid. The algorithm maintains a priority queue of edges on the front and the new starting point for the algorithm is picked as the first element in this queue. The agglomeration algorithm has been developed in Reference [11], and is a variation on the one used by Lallemand et al. [2]. The algorithm is given below:

Pick a starting vertex on a surface element.

Agglomerate control volumes associated with its neighboring vertices which are not already agglomerated.

Define a front as comprised of the exterior faces of the agglomerated control volumes. Place the exposed edges (duals to the exterior faces) in a queue.

Pick the new starting vertex as the unprocessed vertex incident to a new starting edge which is chosen from the following choices given by order of priority:

- An edge on the front that is on the solid wall.
- An edge on the solid wall.
- An edge on the front that is on the far field boundary.
- An edge on the far field boundary.
- The first edge in the queue.

Go to Step 2 until the control volumes for all vertices have been agglomerated.

There are many other ways of choosing the starting vertex in Step 4 of the algorithm, but we have found the above strategy to be the best. The algorithm has been optimized and runs in a time linearly proportional to the number of fine grid vertices.

Although agglomeration is intuitively thought of as grouping fine grid control volumes together to form larger coarse grid cells, an alternate interpretation in terms of a point removal process is also possible. If we consider each agglomerated control volume to be identified by its seed point (i.e. the starting fine grid vertex used to initiate the agglomeration of the cell), then these seed points may be thought of as common to both the fine and coarse levels, while all other fine grid vertices are deleted by the agglomeration procedure in the construction of the coarse level.

In graph theoretical terms, if the initial grid is interpreted as a graph, the agglomeration problem is that of finding a *maximal independent* set with certain desirable properties. A subset of the vertices of a graph is termed an *independent set* if no two vertices in the set are adjacent. An independent set is *maximal* if any vertex not in the set is dominated by (adjacent to) at least one vertex in it. A desirable property for the coarse grids in multigrid is that the number of grid points should decrease by a nearly constant factor when moving from a fine to coarse grid. This factor is 4 in two dimensions, and 8 in three dimensions. The graph problem reduces to finding the maximal independent graph with the minimum cardinality (size) and is nP complete (intractable in polynomial time). However, the heuristic algorithm described above provides a good approximation of this result.

The result of the agglomeration procedure consists of a coarse level set of vertices, as well as a graph of this set (edges joining nearest neighbors), upon which the coarse level discretization is then based. Algebraic multigrid methods employ similar algorithms to determine coarse level variables which are formed as subsets of the fine level variables. However, no coarse level graph is required in the algebraic multigrid method, since the coarse grid operator is determined algebraically. In the agglomeration procedure, the coarse level graphs do not generally form triangulations, thus complicating the matter of discretizing the governing equations of the coarse levels. Another possibility is to neglect the implied agglomeration graph, and simply retriangulate the seed points of the agglomeration process. Although this may not always be possible in the general case (due to boundary effects), this approach has been employed in the following section in order to compare the suitability of the resulting coarse level point sets with those generated by mesh regeneration in the overset mesh multigrid method.

3. FORMULATION OF THE MULTIGRID EQUATIONS

In this section, we formulate the multigrid procedure by examining the convergence rate of a Laplace solver. The multigrid formulation contains three phases: the generation of coarse levels, the construction of the interpolation operators, and the construction of the coarse grid operator. The coarse grid levels are generated by the agglomeration technique which has been described in the previous section. The main concern in this section is the proper formulation of the prolongation and coarse grid operators. In order to isolate the effect of the various operator constructions, as well as the topology of the coarse grids, we restrict ourselves to a two grid system, where the coarse grid is solved exactly at each multigrid cycle. The fine grid is shown in Figure 1, and the coarse agglomerated grid in Figure 2. We compare the efficiency of the agglomeration multigrid approach with that of the independent mesh multigrid approach of [1,4]. The coarse level for this method may either be generated independently using the same grid generation technique employed to generate the fine grid, or by using the agglomeration algorithm as a point removal technique, and then retriangulating the seed points. For this

test case, we were able to reconstruct a triangulation of the seed points which conforms to the coarse level agglomeration graph. While the first approach provides a direct comparison with the previously developed method, the latter removes the effect of the coarse grid topology in the agglomeration technique by ensuring the use of similar coarse grids for both methods.

3.1. Coarse Grid Operator

Since the coarse grid equations cannot be discretized in a straight-forward manner on the agglomerated grid, we resort to a Galerkin coarse grid operator construction, as in the algebraic multigrid case [14]. If we choose the prolongation operator as straight injection (i.e. every constituent fine grid cell of a coarse grid cell is assigned the coarse grid correction value), and volume weighted summation for the restriction operator, the coarse grid operator $\mathbf{R} \mathbf{A} \mathbf{P}$ is equivalent to summing all the discrete equations within each agglomerated cell, and replacing the fine grid variables by the coarse grid variables. Since the discrete operator (for Laplace's equation) is symmetric, all the contributions along edges interior to the agglomerated cell cancel out. Furthermore, since the operator is linear, the contributions of all edges which join two given neighboring agglomerated control volumes can be summed, as shown in Figure 3. Thus the operator $\mathbf{R} \mathbf{A} \mathbf{P}$ results in a nearest neighbor stencil on the coarse agglomerated grid. Furthermore, since the coarse grid matrix entries can be obtained by simple summation, the symmetric and positive properties of the fine grid operator also hold on the coarse grid. In fact, it can be shown that if the fine grid matrix is an M-matrix, the coarse grid matrix will also be an M-matrix [14]. It is interesting to point out that the control-volume approach of discretizing the Euler equations on the agglomerated meshes described in [2,3,10,11] is identical to the Galerkin coarse grid construction described here, provided the non linearities of the Euler equations are handled appropriately.

Table 1 compares the convergence rate obtained by this method on a two-grid agglomerated system, with that of a two grid unrelated mesh multigrid approach, using a coarse triangular grid generated independently, and a coarse triangular grid based on the seed points of the agglomeration algorithm. For all tabulated results, a multigrid V-cycle is employed, with 3 Jacobi pre- and post-smoothing sweeps on the fine grid, and 200 sweeps on the coarse grid (in order to fully converge the coarse grid equations of the two-grid system). In both cases, the convergence rates are much faster than that achieved with the Galerkin coarse grid operator. This degradation of convergence may be due to the different coarse grid operator, or the prolongation and restriction operators (which are taken as linear interpolation in the overset grid method). The effect of the relative "quality" of the coarse grid can be assessed by the difference in convergence rates between the overset grid method using an independent coarse grid, and using the triangulated agglomeration grid. These differences are rather small thus demonstrating the suitability of the agglomerated grid.

The main problem with the above formulation is that the accuracy of the transfer operators is insufficient to guarantee efficient convergence rates. A necessary relation for ensuring multigrid efficiency is given by [15]:

$$m_P + m_R > m \quad (6)$$

where m_P and m_R are defined as the highest degree plus one of the polynomials that are interpolated exactly by \mathbf{P} and \mathbf{R} respectively, and m is the order of the partial differential equation to be solved. In this case, injection is used for the transfer operators, thus m_P and m_R are both equal to one. Since the order of Laplace's equation is 2, the strict inequality is not satisfied. It

is interesting to note that in the case of the convection equation (or the Euler equations) the strict inequality is satisfied since the order of the equations is 1, rather than 2, thus explaining the success of the control-volume formulation of the coarse grid equations for inviscid problems using agglomeration multigrid.

The accuracy of the restriction and/or prolongation operators must therefore be increased. This will affect both the transfer operators themselves, and the coarse grid operator. Rather than strictly adhering to the construction given by equation (4), which can become considerably involved for more complex interpolation operators, we seek a simplified coarse grid operator by examining a one dimensional example. The discretization of a Poisson equation on a one dimensional grid yields the discrete equation:

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = f \quad (7)$$

If a coarse grid is constructed by agglomerating neighboring pairs of cells, as shown in Figure 4, the restriction operator based on injection reads:

$$\bar{r}_B = r_i + r_{i-1} \quad (8)$$

where \bar{r} represents the coarse grid residual, and r_i is the fine grid residual

$$r_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - f \quad (9)$$

The prolongation operator based on injection reads

$$\begin{aligned} u_{i+2} &= u_{i+1} = \bar{u}_A \\ u_i &= u_{i-1} = \bar{u}_B \\ u_{i-1} &= u_{i-2} = \bar{u}_C \end{aligned} \quad (10)$$

where the overbar indicates coarse grid values. The discrete coarse grid equation at B obtained from the application of the Galerkin coarse grid operator becomes

$$\frac{\bar{u}_A - 2\bar{u}_B + \bar{u}_C}{2h^2} = f \quad (11)$$

This obviously results in an inconsistency with the fine grid discretization, for if we were to directly discretize the Poisson equation on the coarse grid we obtain

$$\frac{\bar{u}_A - 2\bar{u}_B + \bar{u}_C}{4h^2} = f \quad (12)$$

Hence, the left-hand sides of equations (11) and (12) differ by a factor of 2. This inconsistency is entirely due to the use of an inadequate prolongation operator. If we use linear interpolation for the prolongation operator, i.e.

$$\begin{aligned} u_{i+1} &= \frac{3}{4}\bar{u}_A + \frac{1}{4}\bar{u}_B \\ u_i &= \frac{1}{4}\bar{u}_A + \frac{3}{4}\bar{u}_B \\ u_{i-1} &= \frac{3}{4}\bar{u}_B + \frac{1}{4}\bar{u}_C \end{aligned} \quad (13)$$

but retain injection for the restriction operator it can be verified that equation (12) is recovered

for the resulting Galerkin coarse grid operator. Note also that the inequality of equation (6) is satisfied for this case. This one-dimensional example suggests a simple fix for the multi-dimensional Galerkin coarse grid operator using injection. We replace the operator $\mathbf{R} \mathbf{A} \mathbf{P}$ with

$$\bar{\mathbf{A}} = \frac{\mathbf{R} \mathbf{A} \mathbf{P}}{2^{n-1}} \quad (14)$$

where $n=2,3,\dots,k$ represents the coarse grid levels. In Table 1, this modified or scaled coarse grid operator can be seen to show significant improvement in convergence rate over the original coarse grid operator for the two grid system. A similar result has been proposed in reference [13].

3.2. Prolongation Operator

While scaling of the coarse grid operator improves the convergence rate of the algorithm, the efficiency still lags that of the overset grid method. This is due to the interpolation operators which are still based on injection, resulting in the violation of inequality (6). When linear interpolation is employed for the restriction and prolongation operators, the agglomeration multigrid convergence rate becomes comparable to that achieved with the overset grid multigrid method, as seen in Table 2. However, linear interpolation operators are not easily constructed on agglomeration meshes. For the test case shown in Table 2, the linear interpolation operators are those employed by the overset grid method operating on a triangulated version of the coarse agglomerated grid (using the seed points for the basis of the triangulation). In general, a triangulation of the agglomerated grid seed points which preserves the boundary may not exist, and this method therefore cannot be generalized. However, this example serves to illustrate the benefits of increasing the accuracy of the interpolation operators. Furthermore, if we employ injection for the restriction operator, but linear interpolation for the prolongation operator, the convergence rate degrades only slightly, as shown in Table 2. This is not surprising, since in this case the inequality of equation (6) is still satisfied.

Thus, a more sophisticated prolongation operator is required in the agglomeration strategy. In order to construct such an operator, we make use of the criterion used in algebraic multigrid methods which states that the prolonged corrections should be algebraically smooth on the fine grid (cf. equation 2). If the coarse agglomerated control volumes are assumed to be represented by their seed points, then these points may be interpreted as common to both coarse and fine grid levels. The appropriate prolongation at these points is thus injection. At the vertices of the fine mesh which are not common to the coarse mesh, rather than using straight injection as in the previous case, we require equation (2) to hold. These equations can be then solved by iterative means (i.e multiple Jacobi iterations). These iterations are similar to those of the base fine grid solver, since the same operator is involved. However the appropriate boundary condition in this case is

$$u_i = \bar{u}_i \quad \text{for } i = \text{seed point} \quad (15)$$

where u_i represents the fine grid corrections at seed points, and \bar{u}_i represents the coarse grid corrections. The fact that the same iterative solver as the base fine grid solver can be employed makes this implicit prolongation operator simple to construct. The application of equation (15) as a boundary condition ensures rapid convergence of any simple iterative scheme, since in general each fine grid point which is not a seed point will be surrounded by seed points which are only one or two neighbor distances away. Since the seed points constitute a maximal independent set of the fine grid, each fine grid point is either a seed point, or a

neighbor of a seed point. For Laplace's equation, this implicit prolongation operator preserves a linear distribution exactly, and closely approximates the prolongation obtained by triangulating the seed points and using linear interpolation, as described above. (Since different triangulations of the seed points lead to different linear interpolation operators, the two cannot be identical).

In Table 2 the convergence rates of the overset grid method using the triangulated seed points as a coarse grid, the agglomeration method using the same linear interpolation prolongation operator as the previous method, and the agglomeration method using the implicit prolongation operator are depicted. Clearly, the implicit prolongation operator produces nearly identical results to the same scheme using the linear interpolation prolongation operator. Furthermore, the overall multigrid convergence rate degrades only slightly when decreasing the number of iterations used to solve the implicit prolongation operator from 50 to 2, supporting the claim that these equations converge rapidly.

3.3. Alternate Coarse Grid Operators

This algorithm is still somewhat slower than the overset grid method. Tables 1 and 2 can be used to assess the relative effects of the various forms of the restriction, prolongation and coarse grid operators. Apparently, the scaled Galerkin coarse grid operator is not as effective as the coarse grid operator obtained by rediscretizing the governing equation on the coarse grid in the overset grid method. An alternative would be to investigate the use of a Galerkin coarse grid operator $R A P$ where P is the implicit prolongation operator described above. However, due to the implicit form of P , this construction is not straight-forward. An explicit form of the prolongation operator could be constructed in a preprocessing phase by (approximately) inverting the matrix which corresponds to the system of linear equations (2). It would also be necessary to set R equal to the transpose of P in the construction of the Galerkin coarse grid operator, in order to preserve the symmetric M-matrix property of the fine grid operator [14]. Rather than follow this route, we construct a Galerkin coarse grid operator using linear interpolation for the prolongation and restriction operators. The linear interpolation operator itself is obtained by triangulating the coarse grid seed points, and the coarse grid operator is constructed algebraically using equation (4). Although such a construction may not be possible in the general case, it is simple to perform and will be used to assess the behavior of the more general construction using the implicit prolongation operator. The convergence rates of the overset grid method, the agglomeration method using the scaled Galerkin coarse grid operator described above, and the Galerkin coarse grid operator based on linear interpolation are compared in Table 3. Clearly, the Galerkin coarse grid operator based on linear interpolation is much more effective than even the geometric coarse grid operator. This suggests that more effective coarse grid operators can be constructed. However, even though this scheme contains the same number of coarse grid points (or coarse level variables), the matrix of the coarse grid operator is no longer based on the implied agglomeration graph, and is much denser. This is the result of the operator no longer relying exclusively on nearest neighbor stencils. Coarse grid evaluations are over three times more expensive than in the other two cases. Thus, this method is not practical, particularly for multi-level applications. (It should be noted that in [14] alternate strategies for selecting coarse grid vertices are exploited to reduce the complexity of the coarse grid operator, but this has not been investigated in this work).

4. APPLICATION TO THE NAVIER-STOKES EQUATIONS

4.1. Base Solver

This approach is next applied to the Reynolds-averaged Navier-Stokes equations. The single grid solution technique is based on a Galerkin finite-element discretization, where the flow variables are assumed to vary linearly over the triangular elements of the mesh. This formulation provides an elegant framework for discretizing both the inviscid and viscous terms of the Navier-Stokes equations. For the inviscid terms, the identical discrete equations can be derived using a control volume analysis where the control volume for a vertex is taken as the cell of the dual mesh surrounding the vertex (cf. Figure 5). Additional artificial dissipation is added as a blend of a laplacian and biharmonic operator, to capture shocks and maintain stability in smooth regions of the flow, respectively. The resulting spatially discretized equations are then integrated in time to obtain the steady-state solution. This is achieved using a multi-stage Runge-Kutta time-stepping scheme. Convergence is accelerated through the use of implicit residual averaging. On the coarse grids, a first-order accurate discretization is employed, since this results in a nearest neighbor stencil, and does not affect the accuracy of the final solution. Further details on the solver can be found in [16].

The single equation turbulence model of Spalart and Allmaras [17] is employed to account for turbulence effects. The turbulence equation is discretized using first-order upwinding on the convective terms, and second-order Galerkin finite-elements on the diffusive and source terms. The turbulence equation is solved simultaneously but decoupled from the flow equations. The convergence of the turbulence equation is also accelerated using the unstructured multigrid technique, thus ensuring similar convergence rates for the flow and turbulence equations, and improving the overall efficiency of the solver.

4.2. Agglomeration Multigrid Strategy

The multigrid strategy employed for the Navier-Stokes equations employs the scaled Galerkin coarse grid operator described above, an injection restriction operator, and an implicit prolongation operator. In order to permit a simple construction of the coarse grid operator, we must be able to express the discretization as an edge-based nearest-neighbor stencil. A first-order accurate discretization of the convective terms is employed on the coarse grids, thus resulting in a nearest-neighbor stencil. In order to employ a similar technique for the viscous terms, we must first be able to express the viscous terms as a series of edge-based flux contributions, rather than as a sequence of two operations (one for the computation of first derivatives, and the other for the second derivative evaluation). This is possible, since the viscous terms are known to result in a nearest neighbor stencil for triangular or tetrahedral meshes, and have been derived previously [18,19]. Hence, once the viscous terms have been expressed as a set of edge based fluxes, the coarse grid equations can be constructed simply by summing and rescaling the equations from the fine grid control volumes which are contained in a given coarse grid control volume. In practice, both inviscid terms and viscous terms are represented as a set of edge-based coefficients on the fine grid multiplying the local flow variables at either end of the edge to form the appropriate flux. The coefficients for the coarse grid operator are thus obtained by dropping all coefficients for edges which are interior to coarse grid agglomerated cells, and summing all coefficients for edges which border on identical coarse grid cells (cf. Figure 3)

The implicit prolongation operator is implemented identically to the method described for the Laplace equation solver. For each coarse agglomerated cell, a fine grid vertex is identified (in this case the seed point of the agglomeration algorithm), which will receive the injected coarse grid correction. The corrections at the remaining fine grid points are computed by performing a specified number of Runge-Kutta iterations of the fine grid governing equations, while holding the seed point values fixed. This implicit prolongation construction can be very advantageous for problems where a maximum principle or a positivity property is required, such as in the turbulence modeling equation (positivity). Injection or even linear interpolation prolongations cannot guarantee such properties, particularly for non-linear equations. However, if the fine grid equations and the coarse grid equations exhibit a maximum or positivity principle, then the prolonged corrected values will also obey the same principle. This is easily seen since the seed points inherit the property from the coarse grid solution, while the other fine grid points receive corrections generated using fine grid iterations. It is also interesting to note that for hyperbolic problems, many modifications to the simple linear interpolation prolongation operator have been suggested in order to account for the hyperbolic nature of the problem [6,20]. The present implicit formulation should presumably take such effects into account automatically. The key to the utility of this approach depends on the relative cost of solving the implicit equations generated by this form of the prolongation operator.

4.3. Results

The first test case involves turbulent flow over a single airfoil. The geometry consists of an RAE2822 airfoil, with a freestream Mach number of 0.73, a Reynolds number of 6.5 million, and a flow incidence of 2.31 degrees. In this case, the Reynolds-averaged Navier-Stokes equations are solved, and turbulence is modeled using the one equation model of Spalart-Allmaras [17]. The mesh employed for this computation contains a total of 18,840 points, and is depicted in Figure 6. The mesh spacing on the airfoil surface is 10^{-5} chords, resulting in stretchings of the order of 500 to 1 in this region. Four grid levels were used in the multigrid algorithm, two of which are depicted in Figure 7. The final converged solution is displayed in terms of computed Mach contours in Figure 8. The identical solution is obtained by all methods discussed here, since the fine grid discretization is unaltered. The convergence histories of the and the agglomerated multigrid method, and the non-nested multigrid method also using four levels, are compared in Figure 9. The agglomerated multigrid algorithm using injection for the prolongation operator achieves a reduction of 6 orders of magnitude in the residual over 300 cycles. When the implicit prolongation operator is employed, using 10 sub-iterations, the residuals are reduced by an additional 1.5 orders of magnitude. Larger numbers of sub-iterations were not found to appreciably affect convergence, indicating that the implicit prolongation equations are adequately converged. When only 2 sub-iterations are employed, the overall convergence rate decreases only slightly, as can be seen from the figure. In terms of CPU time, the strategy involving 10 sub-iterations is clearly not practical, but serves to illustrate the maximum potential benefit afforded by the implicit prolongation operator. The strategy using 2 sub-iterations requires 30% more CPU time per cycle than the direct injection agglomeration multigrid approach and thus achieves approximately the same overall efficiency as the injection approach. The convergence rate of the non-nested multigrid approach appears to be roughly equivalent to that of the agglomeration multigrid run using injection. Furthermore, both methods require approximately the same amount of CPU time per cycle (2 seconds per cycle on a CRAY-YMP-1), and thus are equivalent in overall efficiency. In general, the relative performance of the two methods is somewhat case dependent, and may be influenced

by the construction of the coarse levels in both methods.

The final test case involves the computation of turbulent flow over a three element airfoil. The mesh employed for this case is shown in Figure 10. The total number of mesh points is 55845, and the first point off the airfoil surfaces is placed at a distance of 10^{-6} chords in the normal direction, resulting in aspect ratios of 1000 to 1 in these regions. This level of mesh resolution has been determined as the minimum required for adequate performance prediction of high-lift multi-element airfoil flows [21]. Turbulence is modeled using the one equation model of Spalart-Allmaras [17]. The freestream conditions for this case are: Mach number = 0.2, Reynolds number = 9 million, and an incidence of 16 degrees. The solution in terms of computed Mach contours is qualitatively depicted in Figure 11. Extensive comparison of this solution with experimental data has been reported in [21]. In Figure 12, the convergence rates of the agglomeration and non-nested multigrid strategies are compared. The non-nested multigrid method achieves a residual reduction of almost five orders of magnitude over 400 cycles, while the agglomeration strategy, using injection for the prolongation operator, results in a slightly higher residual after 400 cycles. However, the asymptotic convergence rates of both methods appear to be similar. The non-nested multigrid run required 5.0 seconds per cycle on a CRAY-YMP-1, while the agglomeration approach required 5.8 seconds per cycle on the same machine.

5. CONCLUSIONS

The extension of agglomeration multigrid to viscous flows is certainly not trivial. By experimenting with a simple model equation we have developed a plausible construction for the coarse grid operator and the prolongation operator. In doing so, we have gradually shifted from the somewhat heuristic approach of rediscretizing the governing equations on the coarse agglomerated grids, which succeeds so well for the Euler equations, to an approach more firmly rooted in algebraic multigrid ideas.

While the proposed implicit prolongation operator has been shown to accelerate convergence, overall efficiency depends on the ability to inexpensively solve the resulting implicit equations. Future work will investigate more efficient methods of approximately solving these equations. Our experiments have also shown the possibility of constructing more effective coarse grid operators. Here again, work is required to develop an alternate form of the operator whose cost does not outweigh the afforded gains in overall convergence.

Finally, the performance of the agglomeration algorithm, even without the implicit prolongation operator, appears to be comparable to that of the non-nested multigrid method for the Navier-Stokes cases presented here. This may seem surprising, given the results obtained for Laplace's equation. However, the main factor impeding convergence for both methods in these cases is known to be the effect of grid stretching. We expect the largest efficiency gains to be obtained by relieving this effect through the use of techniques similar to semi-coarsening. These may be implemented by modifying the agglomeration algorithm either based on cell aspect ratios, or on the operator matrix coefficients, thus resulting in the consideration of weighted graphs.

REFERENCES

1. Mavriplis, D. J., "Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes", *AIAA Journal*, Vol 26, No. 7, July 1988, pp. 824-831

2. Lallemand, M. H., Dervieux, A., A Multigrid Finite-Element Method for Solving the Two-Dimensional Euler Equations", *Proceedings of the Third Copper Mountain Conference on Multigrid Methods, Lecture Notes in Pure and Applied Mathematics*, Ed S. F. McCormick, Marcel Dekker Inc., April 1987, pp. 337-363.
3. Smith, W. A., "Multigrid Solution of Transonic Flow on Unstructured Grids", *Recent Advances and Applications in Computational Fluid Dynamics, Proceedings of the ASME Winter Annual Meeting*, Ed. O. Baysal, November 1990.
4. Mavriplis, D. J., "Three Dimensional Unstructured Multigrid for the Euler Equations", *AIAA Journal Vol 30, No 7*, pp. 1753-1761, July 1992.
5. Peraire, J., Peiro, J., and Morgan, K., "A 3D Finite-Element Multigrid Solver for the Euler Equations", *AIAA Paper 92-0449*, January 1992
6. Leclercq, M. P., "Resolution des Equations d'Euler par des Methodes Multigrilles Conditions aux Limites en Regime Hypersonique", *Ph.D Thesis, Applied Math, Universite de Saint-Etienne*, April, 1990.
7. Perez, E., "Finite Element and Multigrid Solution of the Two-Dimensional Euler Equations on a Non-Structured Mesh", *INRIA Report No. 442*, September, 1985.
8. Connell, S. D., and Holmes, D. G., "A 3D Unstructured Adaptive Multigrid Scheme for the Euler Equations", *AIAA Paper 93-3339-CP*, July 1993.
9. Guillard, H., "Node Nested Multigrid with Delaunay Coarsening", *INRIA Report No. 1898*, 1993
10. Venkatakrishnan, V. and Mavriplis, D., "Unstructured Multigrid through Agglomeration", *Proceedings of the 6th Copper Mountain Multigrid Conference, NASA CP 3224*, April 1993
11. Venkatakrishnan, V. and Mavriplis, D., "Agglomeration Multigrid for the 3D Euler Equations" *AIAA Paper 94-0069* January 1994
12. Lallemand, M. H., Steve, H., and Dervieux, A., "Unstructured Multigrid by Volume Agglomeration: Current Status", *Computers and Fluids*, Vol 21 pp. 397-433, 1992.
13. Koobus, B., Lallemand, M. H., and Dervieux, A., "Unstructured Volume-Agglomeration MG: Solution of the Poisson Equation" *INRIA Report 1946*, June 1993
14. Ruge, J. W., and Stuben, K., *Algebraic Multigrid*, in Multigrid Methods, McCormick, S. F., Editor, SIAM Frontiers in Applied Mathematics, SIAM, Philadelphia, 1987, pp. 73--131.
15. *An Introduction to Multigrid Methods*, P. Wesseling, Pure and Applied Mathematics, John Wiley and Sons, 1991
16. Mavriplis, D. J., "Turbulent Flow Calculations Using Unstructured and Adaptive Meshes", *Int. J. Numer. Methods Fluids*, Vol. 13, No. 9, November 1991, pp. 1131-1152
17. Spalart, P. R., and Allmaras S. R., "A One-Equation Turbulence Model for Aerodynamic Flows", *AIAA Paper 92-0439* January 1992
18. Barth, T. J., "Numerical Aspects of Computing Viscous High-Reynolds Number Flows on Unstructured Meshes", *AIAA Paper 91-0721* January, 1991
19. Venkatakrishnan, V. and Mavriplis, D., "Implicit Solvers for Unstructured Meshes", *Journal of Computational Physics*, Vol 105, No. 1, pp. 83-91, March 1993
20. Koren, B., and Hemker, P. W., "Damped Direction-Dependent Multigrid for Hypersonic Flow Computations", *Applied Numerical Mathematics*, Vol 7, pp. 309-328, 1991.
21. Valarezo, W. O., and Mavriplis, D. J., "Navier-Stokes Applications to High-Lift Airfoil Analysis", *AIAA Paper 93-3534*, *AIAA 11th Applied Aerodynamics Conference, Monterey CA*, August 1993

EFFECT OF COARSE GRID OPERATOR				
COARSE GRID	COARSE GRID OP.	RESTRICTION	PROLONGATION	CONVERGENCE RATE
Independent	Rediscretization	Linear	Linear	0.100
Triangulated Seed Pts	Rediscretization	Linear	Linear	0.125
Agglomerated	Galerkin	Injection	Injection	0.512
Agglomerated	Scaled Galerkin	Injection	Injection	0.254

Table 1
Effect of Coarse Grid Operator

EFFECT OF TRANSFER OPERATORS				
COARSE GRID	COARSE GRID OP.	RESTRICTION	PROLONGATION	CONVERGENCE RATE
Triangulated Seed Pts	Rediscretization	Linear	Linear	0.125
Agglomerated	Scaled Galerkin	Injection	Injection	0.254
Agglomerated	Scaled Galerkin	Linear	Linear	0.159
Agglomerated	Scaled Galerkin	Injection	Linear	0.171
Agglomerated	Scaled Galerkin	Injection	Implicit (50 cycles)	0.177
Agglomerated	Scaled Galerkin	Injection	Implicit (5 cycles)	0.178
Agglomerated	Scaled Galerkin	Injection	Implicit (2 cycles)	0.195

Table 2
Effect of Inter-Grid Transfer Operators

EFFECT OF COARSE GRID OPERATOR				
COARSE GRID	COARSE GRID OP.	RESTRICTION	PROLONGATION	CONVERGENCE
Independent	Rediscretization	Linear	Linear	0.100
Triangulated Seed Pts	Rediscretization	Linear	Linear	0.125
Agglomerated	Scaled Galerkin	Linear	Linear	0.159
Agglomerated	Full Galerkin (Using Lin. Int. for R and P)	Linear	Linear	0.060

Table 3
Effect of Coarse Grid Operator

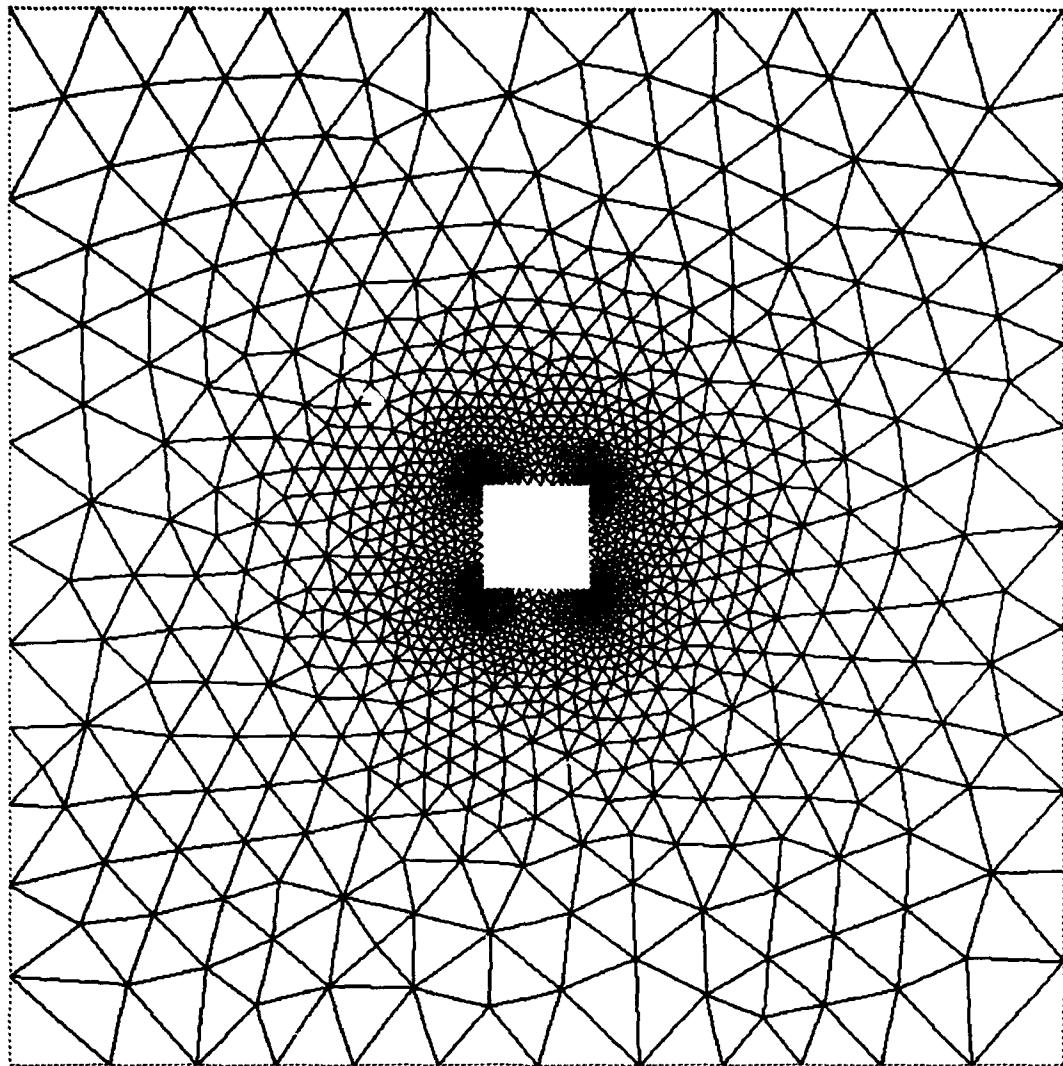


Figure 1
Unstructured Grid Employed For the Solution of Laplace's Equation
(Number of Vertices = 1589)

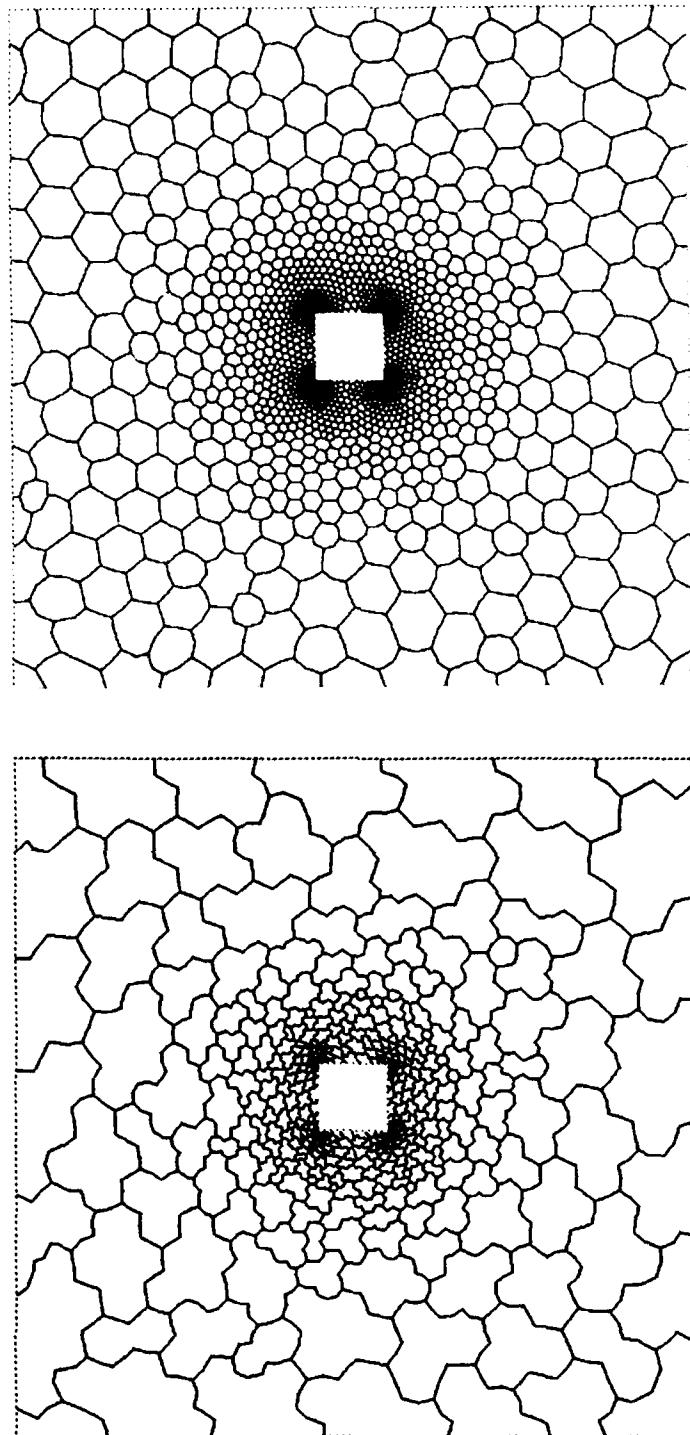


Figure 2
**Dual of Fine Mesh and Coarse Agglomerated Mesh Employed for the Solution
of Laplace's Equation using a Two-Grid Multigrid Procedure**

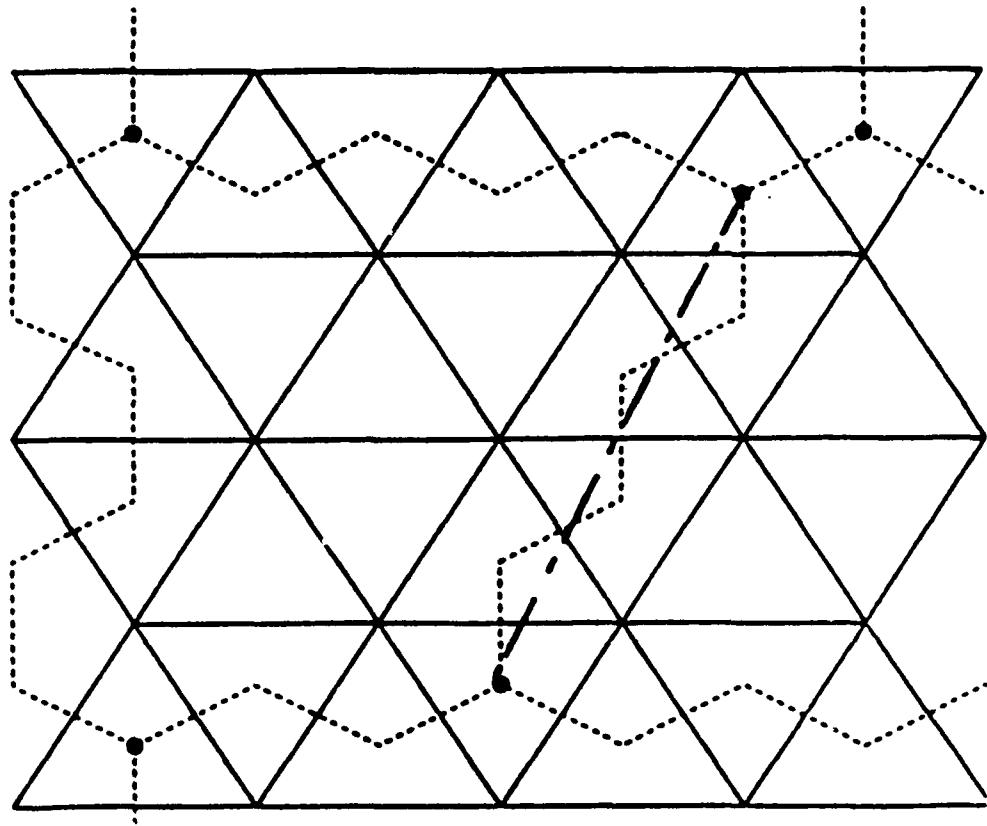


Figure 3
Coarse Agglomerated Control Volume Showing Fine Grid Constituents.
Contributions of Interior Fine Grid Edges Cancel.

Contributions of Edges Joining a Given Neighbor May be Combined into a Single Edge Corresponding to Dashed Line Edge

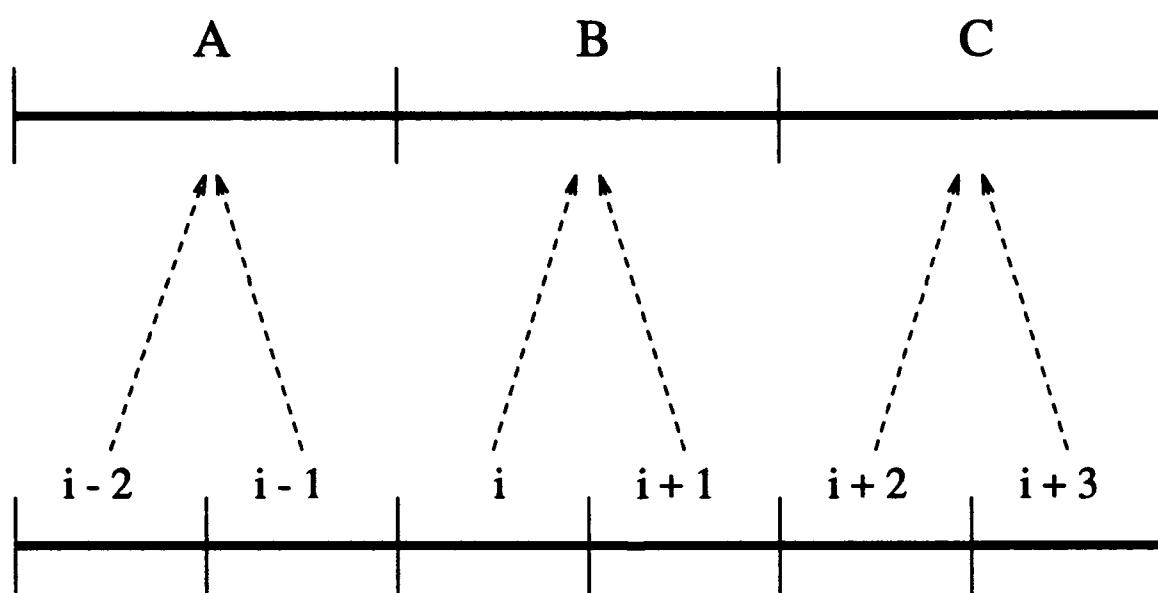


Figure 4
Illustration of Simple 1D Agglomeration Multigrid

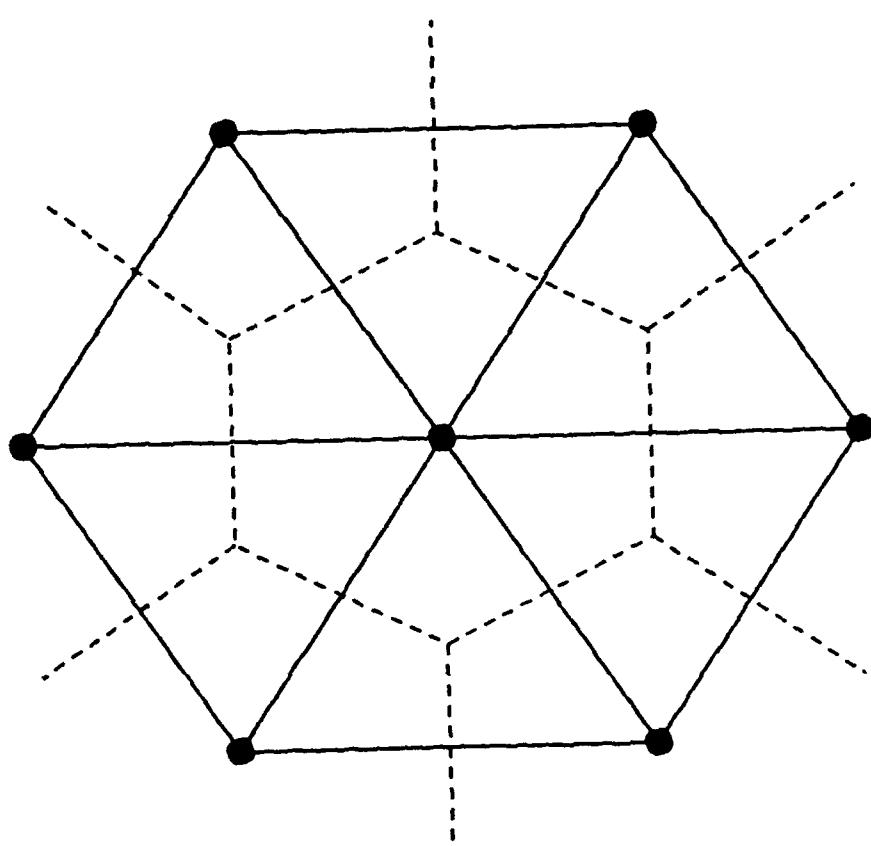


Figure 5
Dual Mesh and Control Volume Employed for Discretization of Navier-Stokes Equations

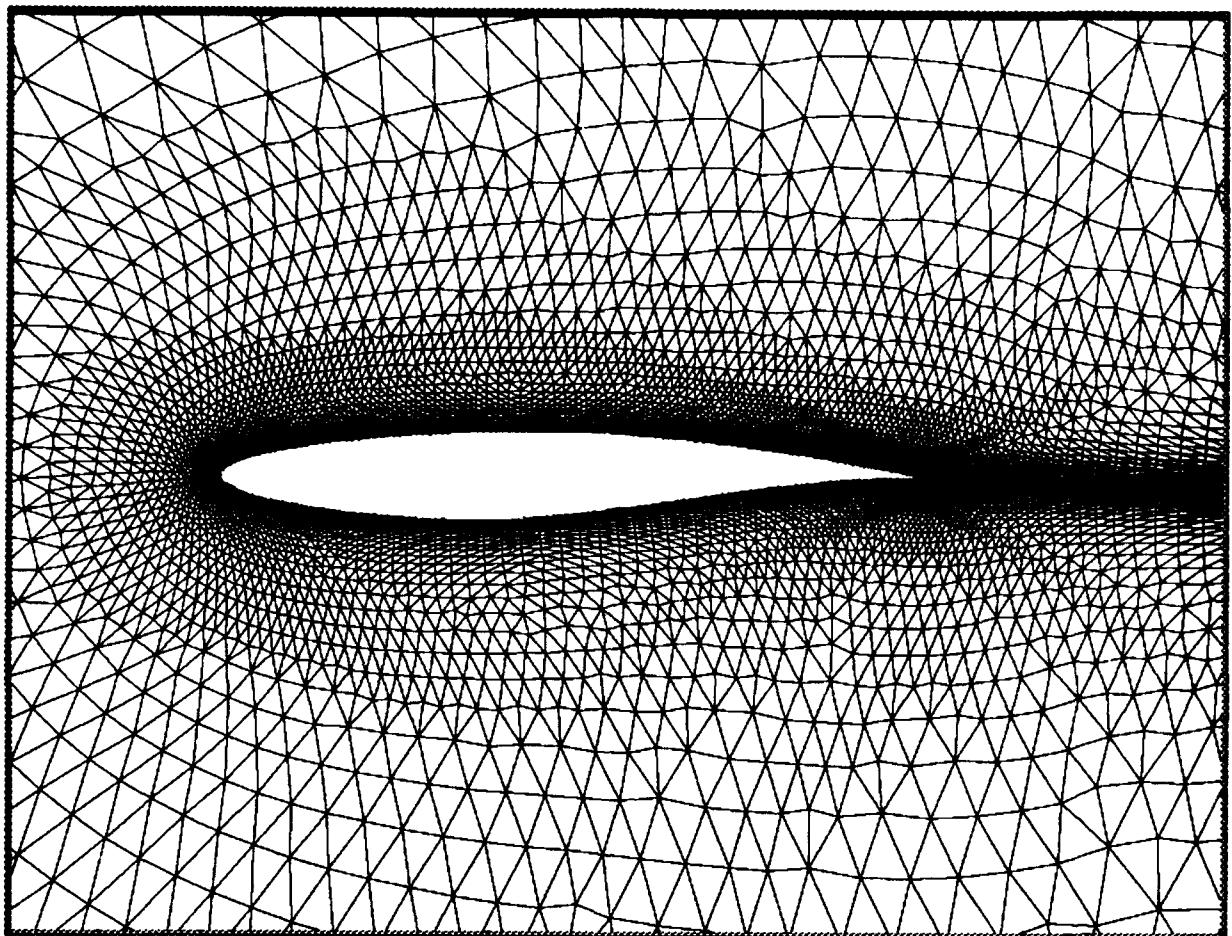


Figure 6
Fine Unstructured Mesh Employed for Computing Viscous Flow over an RAE2822 Airfoil
(Number of Vertices = 18840)

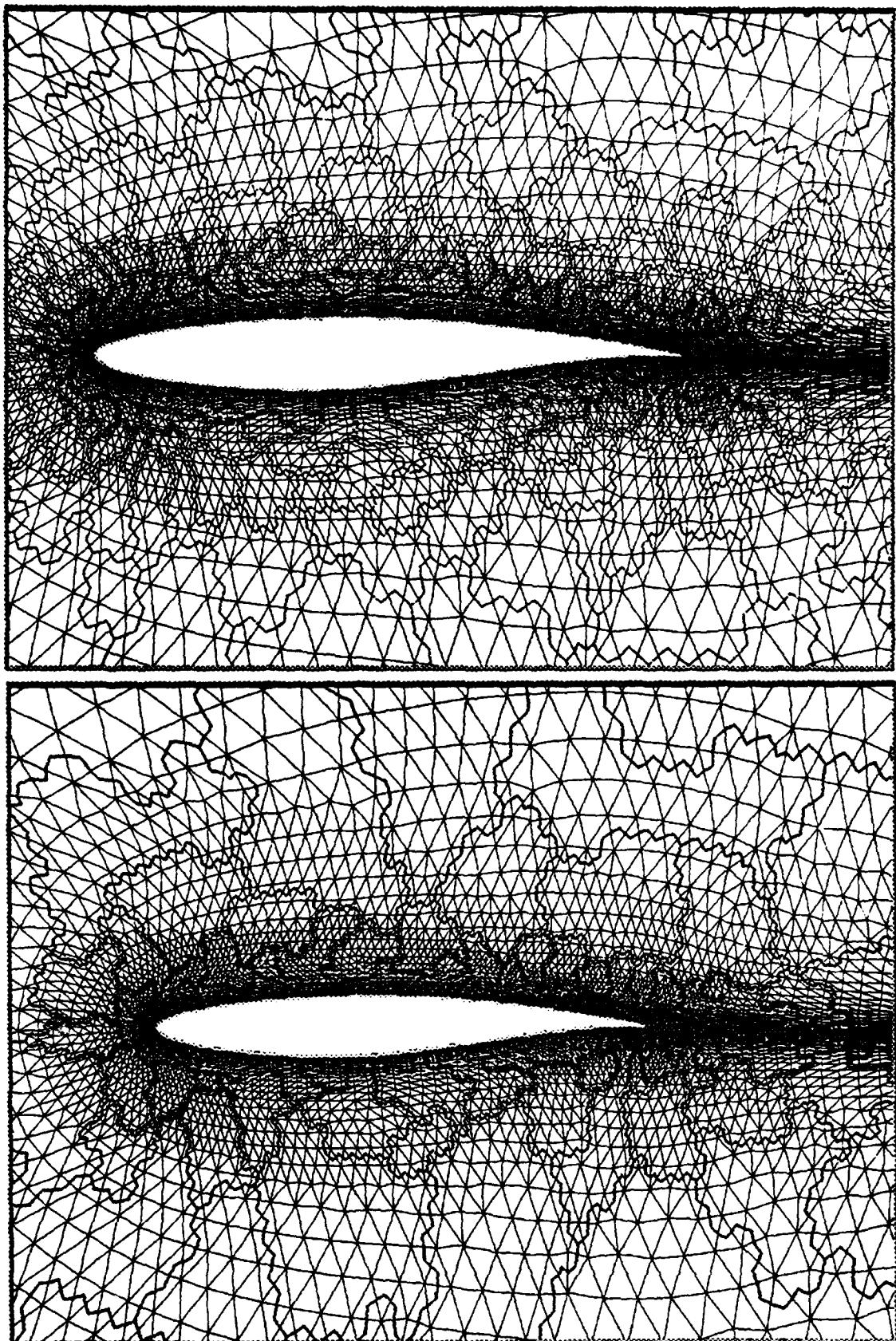


Figure 7
Second and Third Coarse Agglomerated Levels
Employed for Computing Viscous Flow over an RAE2822 Airfoil

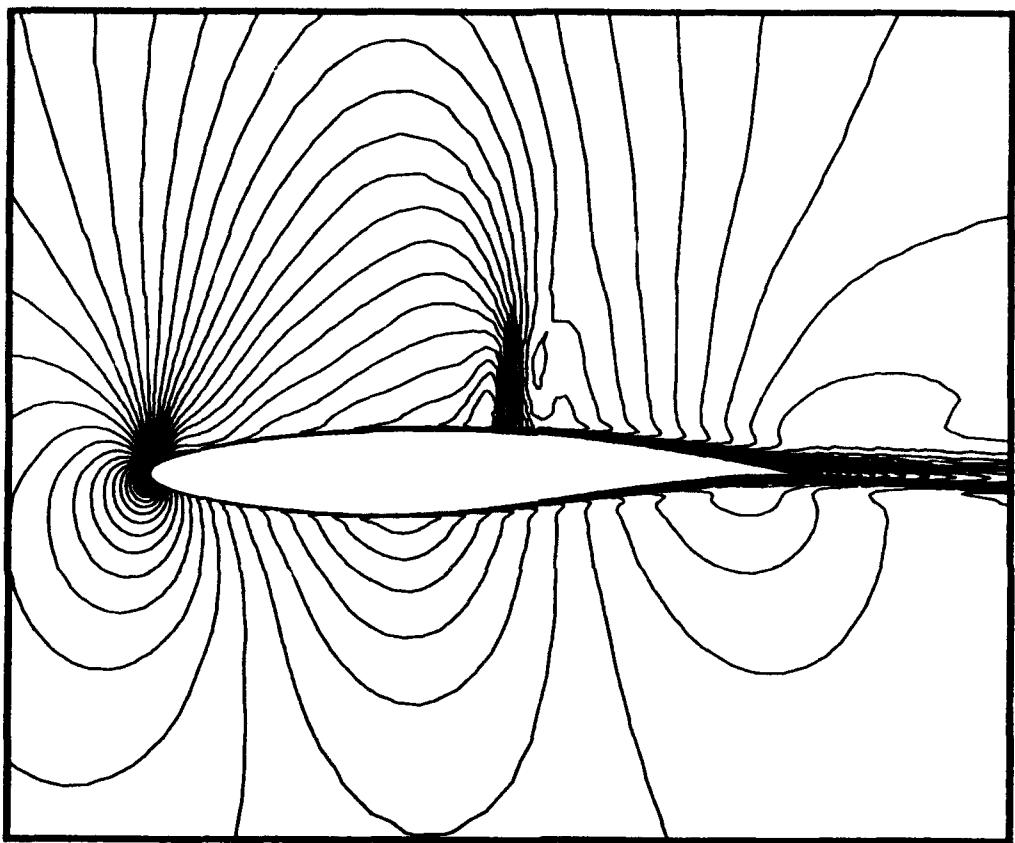


Figure 8

Computed Mach Contours for Viscous Turbulent Flow over an RAE2822 Airfoil
(Mach = 0.73, Incidence = 2.31, Reynolds Number = 6.5 million)

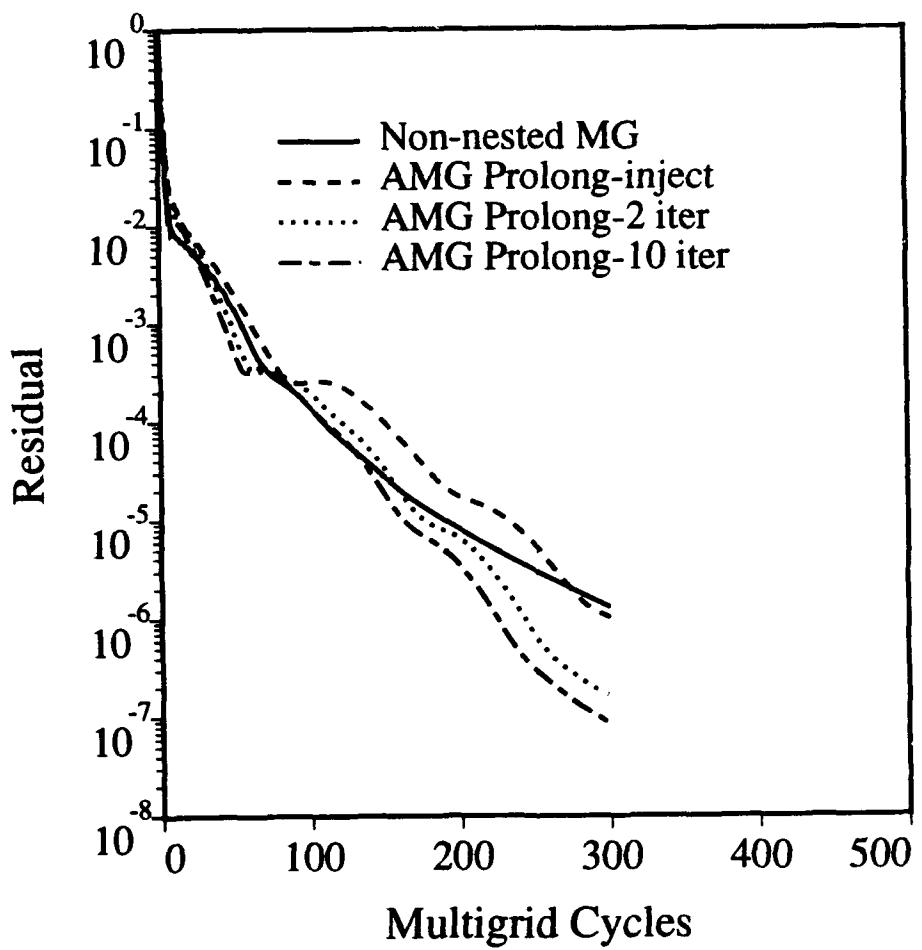


Figure 9
Various Multigrid Convergence Rates Obtained for the Computation of
Viscous Turbulent Flow over an RAE2822 Airfoil

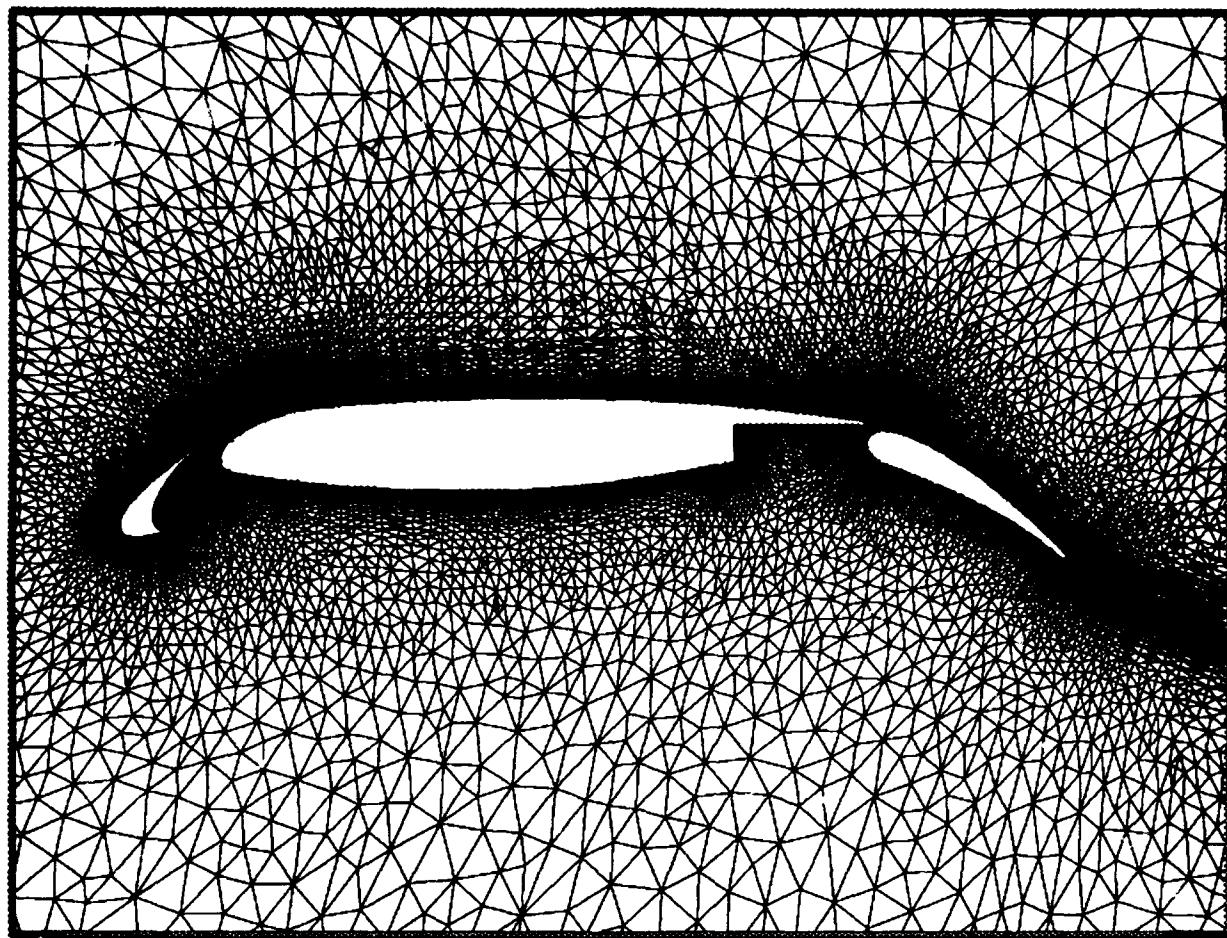


Figure 10

**Fine Unstructured Mesh Employed for Computing Viscous Flow over a Three-Element High-Lift Airfoil Configuration
(Number of Vertices = 55865)**

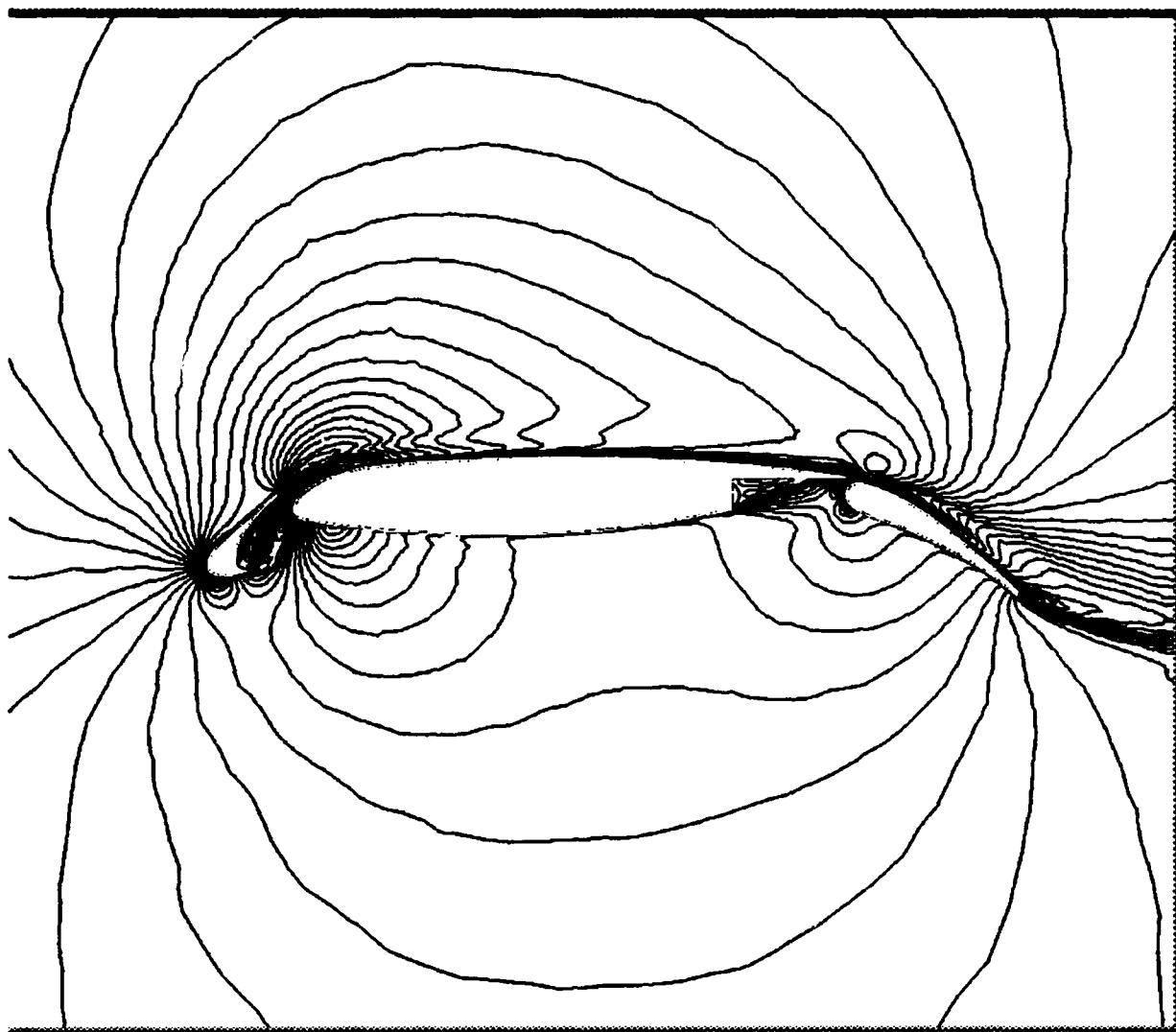


Figure 11
Computed Mach Contours for Turbulent Flow Over Three-Element Airfoil Configuration
(Mach = 0.2, Incidence = 16 degrees, Reynolds Number = 9 million)

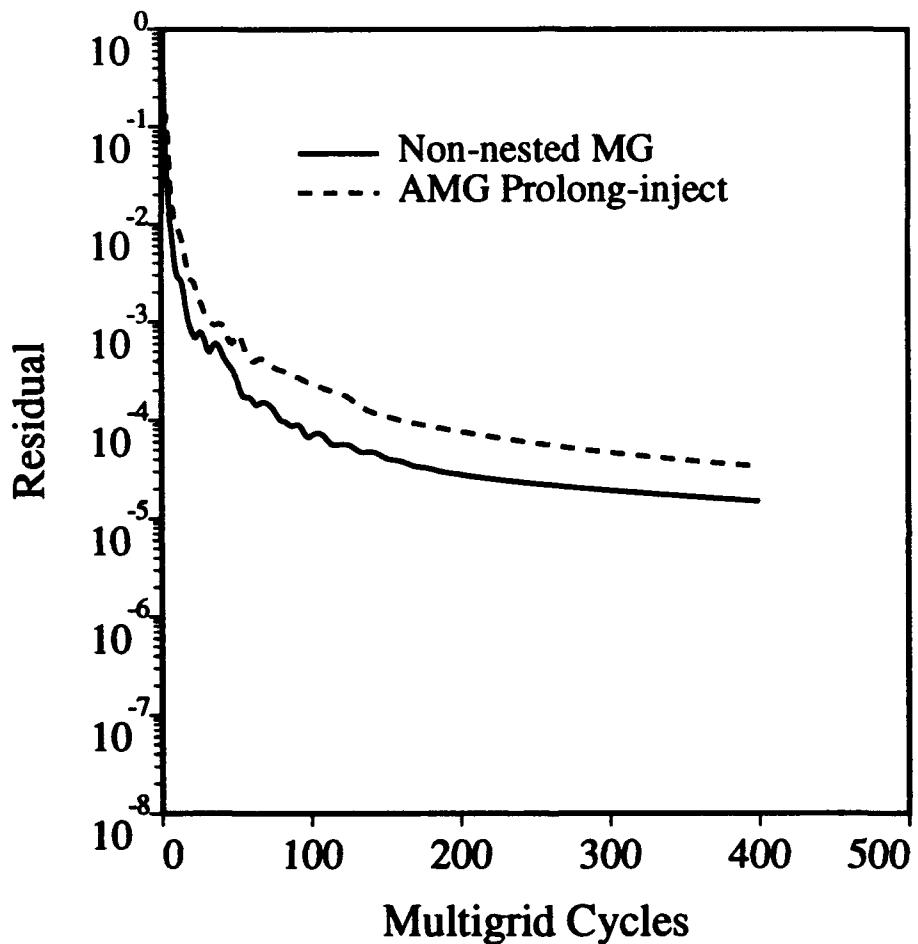


Figure 12
Overset-Mesh and Agglomeration Multigrid Convergence Rates Obtained
for the Computation of Viscous Turbulent Flow
over Three-Element Airfoil Configuration

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</p>			
1. AGENCY USE ONLY(Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
	July 1994	Contractor Report	
4. TITLE AND SUBTITLE		5. FUNDING NUMBERS	
AGGLOMERATION MULTIGRID FOR VISCOUS TURBULENT FLOWS		C NAS1-19480 WU 505-90-52-01	
6. AUTHOR(S)			
D. J. Mavriplis V. Venkatakrishnan			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		8. PERFORMING ORGANIZATION REPORT NUMBER	
Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23681-0001		ICASE Report No. 94-62	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
National Aeronautics and Space Administration Langley Research Center Hampton, VA 23681-0001		NASA CR-194953 ICASE Report No. 94-62	
11. SUPPLEMENTARY NOTES Langley Technical Monitor: Michael F. Card Final Report Submitted to Computers in Fluids Journal!			
12a. DISTRIBUTION/AVAILABILITY STATEMENT		12b. DISTRIBUTION CODE	
Unclassified—Unlimited Subject Category 02, 64			
13. ABSTRACT (Maximum 200 words) Agglomeration multigrid, which has been demonstrated as an efficient and automatic technique for the solution of the Euler equations on unstructured meshes, is extended to viscous turbulent flows. For diffusion terms, coarse grid discretizations are not possible, and more accurate grid transfer operators are required as well. A Galerkin coarse grid operator construction and an implicit prolongation operator are proposed. Their suitability is evaluated by examining their effect on the solution of Laplace's equation. The resulting strategy is employed to solve the Reynolds-averaged Navier-Stokes equations for aerodynamic flows. Convergence rates comparable to those obtained by a previously developed non-nested mesh multigrid approach are demonstrated, and suggestions for further improvements are given.			
14. SUBJECT TERMS		15. NUMBER OF PAGES	
unstructured, multigrid, agglomeration		29	
16. PRICE CODE			
A03			
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		

NSN 7540-01-280-5500

★ U.S. GOVERNMENT PRINTING OFFICE: 1994 - 525-064/23017

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102